

**Amendments to the Claims**

**Please cancel claims 31-45.**

**Please amend claims 1, 3, 4, 6, 15, 16, 18, 19, 21 and 30.**

5 1. (currently amended) ~~A~~ In an electronic device, a method for controlling disposition of a candidate object in an object-oriented environment, said method comprising:

determining a first value indicative of a number of references to said candidate object that are not references from other objects in the object-oriented environment; and

10 determining a second value indicative of a number of references to said candidate object from other objects in the object-oriented environment;

determining a third value indicative of a number of cyclic paths including said candidate object; and

15 controlling disposition of said candidate object on the basis of said first value, said second value and said third value.

2. (original) The method of claim 1, wherein determining a first value comprises reading an external-reference count.

20 3. (currently amended) The method of claim 1, wherein controlling disposition of said candidate object on the basis of said first value comprises:

determining, on the basis of said first value, whether there exists at least one reference to said candidate object that is not from another object in the object-oriented environment; and

marking said candidate object for preservation if there exists at least one reference to said candidate object that is not from another object in the object-oriented environment.

- 5    4. (currently amended)        ~~The method of claim 1, wherein controlling disposition of said candidate object on the basis of said first value further comprises~~ In an electronic device, a method for controlling disposition of a candidate object in an object-oriented environment, said method comprising:

10        determining a second value indicative of a number of references to said candidate object from other objects in the object-oriented environment;

         determining a third value indicative of a number of cyclic paths including said candidate object; and

         controlling disposition of said candidate object on the basis of a predetermined relationship between said second value and said third value.

15

5. (original)    The method of claim 4, wherein determining a second value comprises reading an internal-reference count.

- 20    6. (currently amended)        The method of claim 4, wherein determining a third value comprises:

         identifying a referred object that lies on a path containing a reference originating at said candidate object;

25        determining a fourth value indicative of a number of references to said referred object that originate at other objects in the object-oriented environment, said fourth value being associated with said referred object; and

determining a fifth value indicative of the number of cyclic paths to said candidate object that pass through said referred object, said fifth value being associated with said referred object.

- 5     7. (original) The method of claim 6, wherein determining a third value further comprises:

initializing a sixth value associated with said referred object, said sixth being indicative of a number of cyclic paths known to include said candidate object and said referred object; and

- 10         adjusting said sixth value if said referred object has a reference directly to said candidate object.

8. (original) The method of claim 7, wherein determining a third value further comprises:

identifying a referring object having a reference to said referred object,

- 15         detecting a defined relationship between said fifth value and said sixth value associated with said referred object, and

adjusting a seventh value associated with said referring object in response to detection of said defined relationship, said seventh value being indicative of a number of known cyclic paths that include said candidate object and said referring object.

20

9. (original) The method of claim 8, wherein detecting a defined relationship between said fifth value and said sixth value comprises determining that said fifth value and said sixth value are equal to each other.

10. (original) The method of claim 8, wherein adjusting a seventh value comprises adjusting said seventh value by an amount corresponding to said sixth value.

11. (original) The method of claim 4, wherein controlling disposition of said candidate  
5 object on the basis of said second value and said third value comprises:

determining if said candidate object is externally unreachable or externally reachable;

designating said candidate object for destruction if said candidate object is externally unreachable; and

10 designating said candidate object for preservation if said candidate object is externally reachable.

12. (original) The method of claim 4, wherein:

determining a third value comprises:

15 classifying a path to said candidate object as originating at an external reference or not originating at an external reference; and

controlling disposition of said candidate object comprises designating said candidate object for destruction if no path to said candidate object originates at an external reference.

20

13. (original) The method of claim 12, wherein classifying a path comprises:

for each object on said path, determining if said object has an external reference; and

classifying said path on the basis of whether at least one object on said path has a reference selected from the group consisting of an external reference and an internal reference from an object not reachable from said candidate object.

- 5     14.(original) The method of claim 1, wherein controlling disposition of said candidate object on the basis of said first value comprises:

determining whether said candidate object is referenced from outside of a tree; and

marking said candidate object for preservation if there exists a reference to said candidate object from a tree.

10

15. (currently amended)     ~~A~~ In an electronic device, a method for automatic control of disposition of a candidate object in an object-oriented programming environment, said method comprising:

detecting deletion of a reference to a candidate object;

- 15     determining a number of cyclic paths that include said candidate object in the object-oriented programming environment;

determining a number of internal references to said candidate object, wherein internal references are references from other objects in the object-oriented programming environment; and

- 20     controlling disposition of said candidate object on the basis of a defined relationship between said number of internal references and said number of cyclic paths.

16. (currently amended)     ~~A computer-readable medium~~ computer-readable medium having encoded thereon software for causing a computer to control disposition of a candidate  
25     object in an object-oriented environment, said software comprising instructions for causing a computer to:

determining a first value indicative of a number of references to said candidate object that are not references from other objects in the object-oriented environment; and

5 determining a second value indicative of a number of references to said candidate object from other objects in the object-oriented environment;

determining a third value indicative of a number of cyclic paths including said candidate object; and

controlling disposition of said candidate object on the basis of said first value, said second value and said third value.

10

17. (original) The computer-readable medium of claim 16, wherein said instructions for:

determining a first value comprise instructions for causing said computer to read an external-reference count.

15 18. (currently amended) The computer-readable medium of claim 16, wherein said instructions for controlling disposition of said candidate object on the basis of said first value comprise instructions for causing a computer to:

20 determine, on the basis of said first value, whether there exists at least one reference to said candidate object that is not from another object in the object-oriented environment; and

mark said candidate object for preservation if there exists at least one reference to said candidate object that is not from another object in the object-oriented environment.

25 19. (currently amended) ~~The computer-readable medium of claim 16, wherein said instructions for causing a computer to control disposition of said candidate object on the basis of said first value further comprise instructions~~ A computer-readable medium having encoded

thereon software for causing a computer to control disposition of a candidate object in an object-oriented environment, said software comprising instructions for causing said computer to:

5       determine a second value indicative of a number of references to said candidate object from other objects in the object-oriented environment; and

      determine a third value indicative of a number of cyclic paths including said candidate object; and

      control disposition of said candidate object on the basis of a predetermined relationship between said second value and said third value.

10

20. (original) The computer-readable medium of claim 19, wherein said instructions for causing said computer to determine a second value comprise instructions for causing said computer to read an internal-reference count.

15   21. (currently amended)       The computer-readable medium of claim 19, wherein said instructions for causing said computer to determine a third value comprise instructions for causing said computer to:

      identify a referred object that lies on a path containing a reference originating at said candidate object;

20       determine a fourth value indicative of a number of references to said referred object that originate at other objects in the object-oriented environment, said fourth value being associated with said referred object; and

      determine a fifth value indicative of the number of cyclic paths to said candidate object that pass through said referred object, said fifth value being associated with said referred object.

25

22. (original) The computer-readable medium of claim 21, wherein said instructions for causing said computer to determine a third value further comprise instructions for causing said computer to:

5 initialize a sixth value associated with said referred object, said sixth value being indicative of a number of cyclic paths known to include said candidate object and said referred object; and

adjust said sixth value if said referred object has a reference directly to said candidate object.

10 23. (original) The computer-readable medium of claim 22, wherein said instructions for causing said computer to determine a third value further comprise instructions for causing said computer to:

identify a referring object having a reference to said referred object; and

15 detect a defined relationship between said fifth value and said sixth value associated with said referred object,

adjust a seventh value associated with said referring object in response to detection of said defined relationship, said seventh value being indicative of a number of known cyclic paths that include said candidate object and said referring object.

20 24. (original) The computer-readable medium of claim 23, wherein said instructions for causing said computer to detect a defined relationship between said fifth value and said sixth value comprise instructions for causing said computer to determine that said fifth value and said sixth value are equal to each other.

25 25. (original) The computer-readable medium of claim 23, wherein said instructions for causing said computer to adjust a seventh value comprise instructions for causing said computer to adjust said seventh value by an amount corresponding to said sixth value.



26. (original) The computer-readable medium of claim 19, wherein said instructions for causing computer to control disposition of said candidate object on the basis of said second value and said third value comprise instructions for causing said computer to:

- 5           determine if said candidate object is externally unreachable or externally reachable;  
              designate said candidate object for destruction if said candidate object is externally unreachable; and to  
              designate said candidate object for preservation if said candidate object is externally reachable.

10   27. (original) The computer-readable medium of claim 19, wherein:

              said instructions for causing said computer to determine a third value comprise instructions for causing said computer to:

              classify a path to said candidate object as originating at an external reference or not originating at an external reference; and

- 15           said instructions for causing said computer to control disposition of said candidate object comprise instructions for causing said computer to designate said candidate object for destruction if no path to said candidate object originates at an external reference.

20   28. (original) The computer-readable medium of claim 27, wherein said instructions for causing said computer to classify a path comprise instructions for causing said computer to:

              for each object on said path, determine if said object has an external reference;  
              and

- classify said path on the basis of whether at least one object on said path has reference selected from the group consisting of an external reference and an internal  
25   reference not reachable from said candidate object.

29. (original) The computer-readable medium of claim 16, wherein said instructions for causing said computer to control disposition of said candidate object on the basis of said first value comprise instructions for causing said computer to:

- 5           determine whether said candidate object is referenced from outside of a tree; and
- mark said candidate object for preservation of there exists a reference to said candidate object from a tree.

30. (currently amended)       A computer-readable medium having encoded thereon software
- 10   for causing a computer to automatically control disposition of a candidate object in an object-oriented programming environment, said software comprising instructions for causing said computer to:

            detect deletion of a reference to a candidate object;

            determine a number of cyclic paths that include said candidate object;

- 15           determine a number of internal references to said candidate object, wherein internal references originate from other objects in the object-oriented programming environment;
- and

~~controlling~~ control disposition of said candidate object on the basis of a defined relationship between said number of internal references and said number of cyclic paths.

20

31-45. (canceled)